

⑪ 公開特許公報 (A)

昭63-115004

⑫ Int.CI.

G 01 C 21/00
G 09 B 29/10

識別記号

府内整理番号

N-6666-2F
A-8302-2C

⑬ 公開 昭和63年(1988)5月19日

審査請求 未請求 発明の数 1 (全15頁)

⑭ 発明の名称 車両の現在地推測方法

⑮ 特願 昭61-260420

⑯ 出願 昭61(1986)10月31日

⑰ 発明者 柏崎 隆	埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所沢工場内
⑰ 発明者 安藤 齊	埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所沢工場内
⑰ 発明者 細井 雅幸	埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所沢工場内
⑰ 発明者 福島 敦彦	埼玉県所沢市花園4丁目2610番地 バイオニア株式会社所沢工場内
⑰ 出願人 バイオニア株式会社	
⑰ 代理人 弁理士 藤村 元彦	東京都目黒区目黒1丁目4番1号

明細書

する車両の現在地推測方法。

3. 発明の詳細な説明

技術分野

本発明は、車載ナビゲーション装置における車両の現在地推測方法に関するものである。

背景技術

近年、地図情報をメモリに記憶しておき、その地図情報をメモリから読み出して車両の現在地とともに表示装置に表示させることにより、車両を所定の目的地に誘導する車載ナビゲーション装置が研究、開発されている。

かかるナビゲーション装置では、車両に搭載された走行距離センサや方位センサ等の出力データに基づいて車両の走行距離や方位等を検出し、これに基づいて時々刻々と変化する車両の現在地を推測することにより、ディスプレイに描画されている地図上への現在地の表示が行なわれる。

ところで、道路地図を数値化するに際しては、車両が走行可能な道路のすべてに關して数値化することが望ましいが、センサ類の精度等により、

1. 発明の名称

車両の現在地推測方法

2. 特許請求の範囲

地図の位置情報を数値データとして記憶しており、車両の進行方位を検出する方位センサ及び車両の走行距離を検出する走行距離センサの各出力データに基づいて車両の現在地を推測して地図上に表示し、進行方位の道路データ欄分に対するすれ角及び推測位置から最近傍欄分までの距離がそれぞれ所定範囲内であれば、前記道路データ欄分上に現在地を修正する車載ナビゲーション装置における車両の現在地推測方法であって、前記すれ角又は前記距離が前記所定範囲外になったことを検出し、その地点からの走行距離を算出し、この走行距離に比例した距離に因し、前記最近傍欄分までの距離のサンプリング筋を用いて走行データと道路データの相関をとり、相関性が高ければ前記道路データ上に現在地を修正することを特徴とする。

実施例

すべての道路に同じ正確なナビゲーションを行なうことは困難である。また、数値化する内容も膨大なものとなる。したがって、あるレベルまで数値化を規定せざるを得なく、この場合数値化されていない道路を走行することを想定しなければならない。

発明の属性

本発明は、上述した点に応じたもので、道路地図の数値化のレベルに拘らず、誤りの無いナビゲーションの実現を可能とした車両の現在地推測方法を提供することを目的とする。

本発明による車両の現在地推測方法は、進行方位の道路データ区分に対するずれ角又は加速度から最近傍段分までの距離が所定範囲外になったことを検出し、その地点からの走行距離を算出し、この走行距離に比例した距離に関し、最近傍段分までの距離のサンプリング率を用いて走行データと道路データの相関をとり、相関性がなければ前記道路データ上に現在地を修正することを特徴としている。

U 7 の各種の処理プログラムやその他必要な情報が予め書き込まれた ROM (リード・オンリ・メモリ) 8 と、プログラムを実行する上で必要な情報の書き込み及び読み出しが行なわれる RAM (ランダム・アクセス・メモリ) 9 と、いわゆる CD-ROM、I C カード等からなり、デジタル化 (数値化) された地図情報が記録された記録媒体 10 と、V-RAM (Video RAM) 等からなるグラフィックメモリ 11 と、CPU 7 から送られてくる地図等のグラフィックデータをグラフィックメモリ 11 に描画しあつてして CRT 等のディスプレイ 12 に表示すべく制御するグラフィックコントローラ 13 とから構成されている。入力装置 14 はキーボード等からなり、使用者によるキー入力により各種の指令等をシステムコントローラ 5 に対して発する。

記録媒体 10 には地図情報が記録されるのであるが、そのデータ構造について以下に説明する。まず、第 2 図 (A) に示すように、日本全国を例えば 16384 (-2¹⁴) [m] 四方のメッシュ

以下、本発明の実施例を图に示して詳細に説明する。

第 1 図は、本発明に係る車載ナビゲーション装置の構成を示すブロック図である。同図において、1 は地磁気について車両の方位データを出力するための地磁気センサ、2 は車両の角速度を検出するための角速度センサ、3 は車両の移動距離を検出するための走行距離センサ、4 は速度及び経度緯度から車両の現在地を検出するための GPS (Global Positioning System) 装置であり、これら各センサ (装置) の出力はシステムコントローラ 5 に供給される。

システムコントローラ 5 は、各センサ (装置) 1~4 の出力を入力とし A/D (アナログ/デジタル) 変換等を行なうインターフェース 6 と、柱々の画像データ処理を行なうとともにインターフェース 6 から順次送られてくる各センサ (装置) 1~4 の出力データに基づいて車両の移動量等を算出する CPU (中央処理回路) 7 と、この CP

に分割し、このときの 1 つのメッシュをテリトリーと呼ぶ。テリトリーはテリトリー NO. (Tx, Ty) で識別され、各テリトリーには例えば図の左下のテリトリーを基準にテリトリー NO. が付与される。テリトリー NO. は現在地 (Crx, Crty) より求まる。テリトリーは本データ構造の中で最も大きな管理単位となる。地図データファイル全体の構成が第 2 図 (B) に示されており、テリトリー ID ファイルには、第 2 図 (C) に示すように、テリトリー NO. (Tx, Ty) のファイルにおける先頭アドレス、テリトリーの左下の緯度 (実数)、テリトリーの左下の経度 (実数)、地磁気の偏角 (実数) 等のデータが各テリトリー毎に書き込まれている。

テリトリー ファイルは本データ構造において最も重要なファイルであり、各種の地図データや地図描画に必要なデータが書き込まれている。第 3 図 (A) において、ナビ ID 及びセクションテーブルがナビゲーションにおける道路及び交差点検索用ファイル、ピクチャーリストが表示管理用ファ

イル、道路セクションデータから交差点データまでが実際の地図データである。地図データは、第3図(B)に示すように、階層構造となっており、最下層が川、坂、湖等のポリゴンデータ、その上が道路、林道等のラインデータ、その上が各格マーク等のキャラクタデータ、その上が地名等の文字データ、そして最上層が交差点データとなっている。最上層の交差点データは復述する交差点引込みのために用いられるデータであり、ディスプレイ上には表示されない。

次に、第4図(A)に示すように、1つのテリトリーの中を例えば256分割し、これにより得られる1024(2¹⁰) [m]四方のメッシュをユニットと呼ぶ。このユニットも同様にユニットNO. (Nx, Ny)で管理され、そのNO.

(Nx, Ny)は現在地(Crntx, Crnty)より求まる。ユニットは中間的な管理単位で、地図情報はこの単位で記録され、ユニットが256個集まってテリトリーファイルを構成する。地図描写的際はこの単位を基に行なわれるので、描画の基

本単位とおうことができる。ナビIDファイルには、第4図(B)に示すように、ユニットNO. (Nx, Ny)のファイルにおけるライン先頭アドレス、交差点先頭アドレス、道路セクション先頭アドレス、交差点先頭アドレス等のデータが各ユニット単位に書き込まれている。

更に、第5図(A)に示すように、1つのユニット内を例えれば16分割し、これにより得られる256(2⁸) [m]四方のメッシュをセクションと呼ぶ。このセクションも同様にセクションNO. (Sx, Sy)で管理され、そのNO. (Sx, Sy)は現在地(Crntx, Crnty)より求まる。セクションは最も小さい管理単位であり、この範囲内の部分(部分の聚りで道路等が表わされる)や交差点の情報を第5図(B), (C)に示す如くセクションテーブルとして、更に第6図(A), (B)及び第7図(A), (B)に示す如くセクションデータとしてテリトリーファイルに登録されている。

また、第3図(A)に示すように、テリトリー

ファイル内には表示管理用のピクチャーファイルがある。本実施例では、地図データの縮尺が例えば2.5万分の1, 5万分の1, 10万分の1の3種類に設定されており、実際の地図データとしては、最も縮尺の大きい2.5万分の1のものだけを持っている。各縮尺の地図は、第8図～第10図の各図(A)に示すように、エリアに分割され、このエリアはエリアNO. (Anx, Any)で管理される。エリアNO. (Anx, Any)は現在地(Crntx, Crnty)より求まる。縮尺が2.5万分の1の場合、エリアNO. とユニットNO. は同じであり、5万分の1の場合には1つのエリアがユニットファイル4個分となり、10万分の1の場合には1つのエリアがユニット16個分となる。また、各縮尺のピクチャーファイルには、第8図～第10図の各図(B)にそれぞれ示すように、その縮尺の地図を表示するのに必要なポリゴン、ライン、キャラクタ、文字データの先頭アドレスとデータサイズが記録されている。

次いで、ポリゴンデータとラインデータについ

て説明する。ポリゴンデータとラインデータは、第11図(A)及び第12図(A)に示すように、始点と終点で表わされる聚りのあるベクトル(部分)で表わされる。ここで、最も縮尺の大きい2.5万分の1の地図データで5万分の1や10万分の1の地図を表現すると、始点・終点間が縮まるのでディスプレイ上で見た限りでは、全ての点を表示しなくても差し支えないことがある。このことを考慮に入れて、ディスプレイ上に表示した場合に、見た目上省略しても差し支えない点の情報を、第11図(B)及び第12図(B)に示すように、予めポリゴン及びラインデータの各端引きビットを入れておく。そして、各縮尺の表示時に端引きビットをチェックして必要に応じて端引きビットに情報が入っている点を除く、いわゆる端引きを行なうことにより、表示する部分(ベクトル)数を減らすことができる。

また、第13図(A)に示すように、1ユニット内に存在する交差点の全てに通し番号(xn, yn)が付されている。ところで、交差点には、直交

型、Y字路、5叉路等々あるが、特に方位の似た道路が複数入っている交差点では、この交差点を通過したときに、センサの精度、計算誤差、地図精度等により道路の選択を誤り、ディスプレイ上に現在地が表示されている道路と実際に走行している道路とが一致しない状態が生ずる可能性がある。そこで、このような交差点に対しては、第13図(B)に示すように、交差点の對易度を示す對易度データを交差点データ中の對易度ビットにいれておく。そして、交差点を通過するときは、この對易度データに基づいた処理を行なえば、誤った道路の選択を防止できるのである。

次に、地図データの表示に関して、グラフィックメモリ11として例えばV-RAMを用いた場合について説明する。表示の構成としては、第14図(A)に示すように、512(ドット)×512(ドット)のV-RAM上で画面を16分割し、それぞれのエリアに独立した1枚の地図を表示するようにする。1エリアは128(ドット)×128(ドット)の1ユニットであり、更に1

6分割することにより1エリアは32(ドット)×32(ドット)の1セクションとなる(第14図(B)、(C)を参照)。實際の車載ディスプレイには、第14図(A)の中央の4箇面に相当する256(ドット)×256(ドット)のエリア(太線で囲ったエリア)が表示され、このエリアがV-RAM上を移動することによって車両の現在地の動きを表現する。

次に、CPU7によって実行される基本的な手順を第15図のフローチャートに従って説明する。

CPU7は、先ず最初にプログラムを実行させるためのイニシャライズを行ない(ステップS1)、しかる後車両の現在地がセットされているか否かを判断する(ステップS2)。現在地がセットされていない場合は、現在地セットルーチンの実行(ステップS3)、例えば入力装置14でのキー入力による現在地のセットが行なわれる。次に、走行距離を零にし(ステップS4)、次いで入力装置14からのキー入力が有るか否かの判断を行なう(ステップS5)。

キー入力が無い場合は、ディスプレイ12上に現在地周辺の地図表示を行なうとともに車両の現在位置及びその方位を例えば車両マークにてこの地図上に表示し、車両が移動したらその移動に伴い地図をスクロールさせ、更に現在グラフィックメモリ11上にある地図データの範囲を車両位置が越えそうなときには、記録媒体10から必要な地図データを読み出してディスプレイ12上に表示する(ステップS6)。

キー入力が有ると、その入力データに応じて現在地の再セット(ステップS7)、センサ補正(ステップS8)、目的地セット(ステップS9)及び地図の拡大・縮小(ステップS10)の各ルーチンを実行する。

また、CPU7はタイマーによる割込みにより、第16図に示すように、一定時間間隔で地磁気センサ1及び角速度センサ2の各出力データに基づいて車両の方位を常に計算する処理を行なう(ステップS11、S12)。

CPU7は更に、走行距離センサ3よりデータ

が入力された場合は、走行距離センサによる割込み処理を行なう。この割込み処理では、第17図に示すように、走行距離と方位からの現在地の算出(ステップS13)、右折、左折の判定(ステップS14)、道路への引込み(ステップS15)、交差点引込み(ステップS16)、走行距離による引込み(ステップS17)が実行される。なお、このステップS13～ステップS17における各処理に関しては、後で詳細に説明する。

また、GPS装置4より得られる緯度、経度データは、第18図に示すように、GPSデータ受信割込みにより処理され、現在地データとして座標変換される(ステップS18)。

車両の走行距離は走行距離センサ3の出力から求められる。この走行距離センサ3としては、例えば、古のいわゆるスピードメータケーブルの回転数(JIS規格では、637回転/Km)より1回転の距離を積分することにより走行距離を求める構成のものが用いられるが、センサ3の精度により得られる走行距離に誤差が生ずることは避

けられない。また、センサ3の精度だけではなく、地図の精度、タイヤの空気圧の変化、或はスリップ等も走行距離の誤差の要因となる。従って、走行距離の修正を成々行なわないと、正確に距離を求めることができなくなってしまうことになる。このため、走行距離センサ3の出力より得た実測の距離と地図データより得た距離とから距離補正係数 α_s を求め、この補正係数 α_s を用いて距離補正を行なうことにより、走行距離を常に正確に換出できるのである。

また、車両の方位は地磁気センサ1の出力から求められる。この方位検出方法に関しては、本出版人等による特許昭60-282341号明細書等に記載されている。この地磁気センサ1の示す北は磁北であり、地図北ではない。このため、磁北が地図北に対してもういた場合、第19図に示すように、基準位置から一定距離だけ走行したときに地磁気センサ1の出力から得られる推算現在地 P_1 は実際の現在地 P_2 に対してずれを生じることになる。そのため、地磁気センサ1より求

めた方位を地図方位に変換する作業が必要となる。この変換作業は、第20図に示すように、2次元幾何の座標変換である四元角、即ち方位補正係数 θ_s によって行なわれる。この方位補正係数 θ_s は地域により変化し、更に地磁気センサ1を車体に取り付けたときに生じる取付け誤差によっても変わる。この方位補正係数 θ_s は、第21図に示すように、当該係数を基として位置のわかっている2点間を走行し、慣性航法により求められた現在地と到着点との誤差により求めることができる。この方位補正係数 θ_s を用いて方位補正を行なうことにより、車両の方位を常に正確に検出できるのである。

なお、距離補正係数 α_s 及び方位補正係数 θ_s の算出方法は、本出版人等による特許昭60-282344号明細書等に記載されている。

車両の走行時には、最近傍交差点への引込みや最近傍桟分への引込みが適宜行なわれる所以あるが、この引込みを行なうためには、現在地に最も近い道路（最近傍桟分）や交差点（最近傍交差点）

を探し出す作業が必要となる。この最近傍交差点や最近傍桟分をサーチする作業は、桟分や交差点データの量が多い、即ちサーチエリアが広いと、時間がかかってしまい、時々刻々と変化する現在地をスムーズに表示できないことになる。ところが、本実施例においては、第2図～第5図に示したデータ構造から明らかのように、現在地からのサーチエリアをできるだけ小さくし、かつそのエリアに入る桟分や交差点のデータを管理するデータ（セクションデータ、セクションテーブル）を持たせていることにより、最小単位のセクションをサーチエリアとしてその中から桟分や交差点をサーチすることができるので、サーチに要する時間は短縮できるのである。以下、CPU7によって実行される現在地から最近傍桟分と最近傍交差点をサーチする手順を、第22図のフローチャートに従って説明する。

CPU7は先ず、現在地($Crntx, Crnty$)からテリトリーNO. (T_x, T_y), ユニットNO. (N_x, N_y), セクションNO. ($S_x,$

S_y)をそれぞれ求める（ステップS20～S22）。これは、各エリアが 2^n 単位で分割されているので、簡単な演算（割り）で求めることができます。次に、セクションをサーチエリアとして、この中に存在する桟分と交差点データをセクションテーブルとセクションデータを参照することによりロードする（ステップS23～S25）。ロードしたデータを基に、現在地からサーチエリア内の全ての桟分までの距離（桟分に対する垂線の長さ）、全ての交差点までの距離を計算し、それらを比較することによって最近傍桟分と最近傍交差点を得ることができる（ステップS26）。サーチを行なう際のスピードは、桟分の本数や交差点の個数に比例するが、前述したデータ構造に基づくサーチ方式によれば、サーチエリア（セクション）が小さく、計算の対象となる桟分の本数や交差点の個数が少ないので、高速サーチが可能となるのである。

次いで、CPU7によって実行される桟分引込みルーチン及び交差点引込みルーチンの手順を、

第23図及び第24図のフローチャートに従って説明する。

第23図のね分引込みルーチンは一定走行距離、例えば20[m]点に行なわれる。CPU7は先ず、第22図の手順に従って最近傍ね分しをリーチし、そのね分しについて、現在地からね分しまでの距離 R_{ij} 及びね分しの地図座となす角度 θ_{ij} を求める(ステップS30)。そして自車角度 θ_c と上記角度 θ_{ij} との差の絶対値と所定のスレショルド値 θ_{th} とを比較し(ステップS31)、スレショルド値 θ_{th} よりも大きい場合は、引込みを行なうための有効ね分が存在せず、しかも道路から外れたと見なす(ステップS32)。

スレショルド値 θ_{th} 以下の場合は、次に上記距離 R_{ij} と所定のスレショルド値 R_{th} とを比較し(ステップS33)、距離 R_{ij} がスレショルド値 R_{th} よりも小さいか等しい場合は、そのね分しは有効ね分であり道路上であると見なし(ステップS34)、ね分上に引込みを行なって現在地の修正を行なう(ステップS35)。また、距離 R_{ij}

ルーチンについて、第25図及び第26図のフローチャートに従って説明する。

第25図は、サンプルされたデータを配列にセーブする距離差データセーブルーチンであり、タイマー割込みによって実行される。CPU7は先ず、道路引込み中であるか否かを判断し(ステップS50)、引込み中でなければ、道路から外れてからの走行距離を表わすカウント値 N_{dc} をインクリメントする(ステップS51)。次に、最近傍ね分なる有効ね分が有るか否かを判断し(ステップS52)、有効ね分が有る場合は、最近傍ね分までの距離 $E(i)$ と前回求めた距離 $E(i-1)$ との差分 $D(i)$ を配列にセーブする(ステップS53)。有効ね分が無い場合は、1回目に極性を逆にした大きな数値をダミーデータとして配列にセーブする(ステップS54)。そしてボインタを“1”だけ進め(ステップS55)、かかる後リターンする。

第26図は、バターンマッチングルーチンであり、メインルーチンから所定のタイミングで呼び

がスレショルド値 θ_{th} よりも大きい場合には、有効ね分は存在するが道路から外れているものと判断する(ステップS36)。

一方、第24図の交差点引込みルーチンは交差点フラグがオンのときに行なわれる。CPU7は先ず、第22図の手順に従って最近傍交差点をリーチし、その交差点について交差点までの距離 R_c (ステップS40)。そして距離 R_c と所定のスレショルド値 R_{th} とを比較し(ステップS41)、距離 R_c がスレショルド値 R_{th} よりも小さいか等しい場合は、引込みを行なうべき交差点があると判断し(ステップS42)、その交差点に現在地を修正する(ステップS43)。距離 R_c がスレショルド値 R_{th} よりも大きい場合は、引込みを行なうべき交差点が無いものとして引込みを行わない(ステップS44)。

次に、道路データから外れた所を走行し、再度道路データ近辺を走行した場合、走行データと道路データとの相間をとり、相間性が高ければ道路データに引き込む、いわゆるバターンマッチング

出されるものである。CPU7は先ず、引込み中であるか否かを判断し(ステップS60)、引込み中であれば、マッチングの必要がないので、そのままリターンする。次に、有効ね分が有るか否かを判断し(ステップS61)、有効ね分がなければ、道路データから一定距離以上離れて走っているので、そのままリターンする。続いて、最小単位距離以上走行したか否かを判断し(ステップS62)、走行していない場合は、そのままリターンする。

次に、道路から外れてからの走行距離が一定の距離内であるか否かを判断し(ステップS63)、一定の距離内である場合は、地磁気センサ補正ルーチンを実行する(ステップS64)。この地磁気センサ補正ルーチンでは、両両が例えば読みを通過した際、車両ポーテーが着用を受けると、地磁気センサ1の出力がすべて道路データに乗らなくなる可能性があるので、地磁気センサ1の出力のそれを検出し、それにに基づいて地磁気センサ1の出力データの補正が行なわれる。

ステップS65では、道路から外れてからの走行距離から、マッキングをとる距離の計りが行なわれる。第25図のステップS51で説明したカウンタのカウント回数Nocに単位距離L.uを乗じたものが、道路から外れてからの走行距離Noc・L.uであり、これに係数Kを乗じたものをマッチング距離L.0(-K・Noc・L.u)とする(ステップS65)。本次実験では、係数KをK=0.25とし、またマッチング距離L.0の最大値、最小値を設定して制限している。

ステップS66では、第25図の距離差データセーブルーチンにおいてセーブされたデータについて、最新のデータから以前のマッチング距離分のデータに因し、その標準偏差の及び平均値を求める計りが行なわれる(ステップS66)。そして、この求められた標準偏差のマッチング距離L.0で割ることにより、単位走行距離当たりの道路データ图形と走行軌跡图形との相間を表わす相間係数Es(-0/L.0)を求める(ステップS67)。統いて、相間係数Esを所定の設定値Esthと比

れし(ステップS68)、Es < Esth の場合は、単位走行距離当たりの道路データ图形と走行軌跡图形との相間性が十分に高いと判断し、最近傍移分への引込みを行なうことによる現在地の修正を行なう(ステップS69)。相間性が低い場合は、そのままリターンし、マッチングルーチンを繰り返す。

説明の効率

以上説明したように、本発明によれば、地図データから外れてからの走行距離に応じてマッチング距離を変えることにより、より正確に道路データに引き込むことができ、また道路データの無い所も走行できるため、全ての道路をデータ化しなくとも、正確なナビゲーションを行なうことが可能となるとともに、経済の数値化レベルにも対応できることになる。

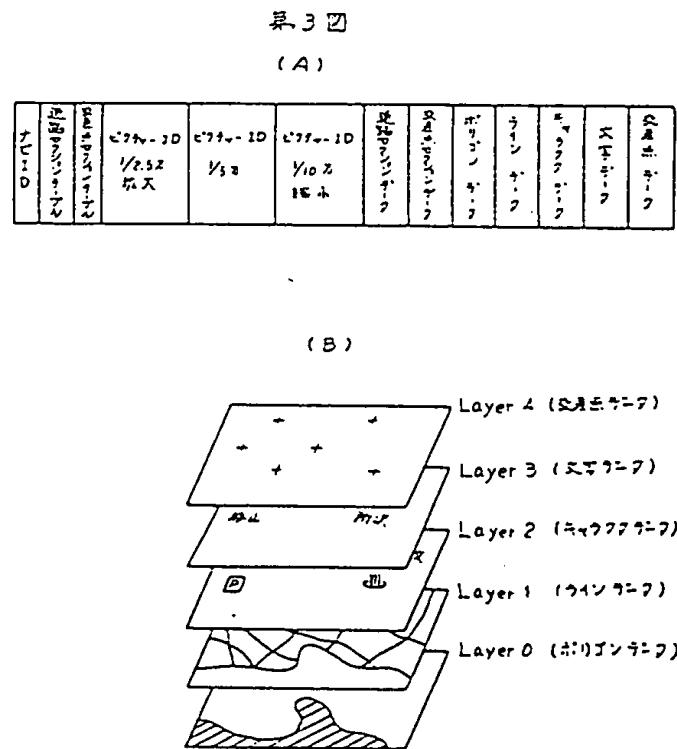
4. 図面の簡略な説明

第1図は本発明による車両の現在地推測方法が適用される車載ナビゲーション装置の構成を示すブロック図、第2図(A)～(C)乃至第13図

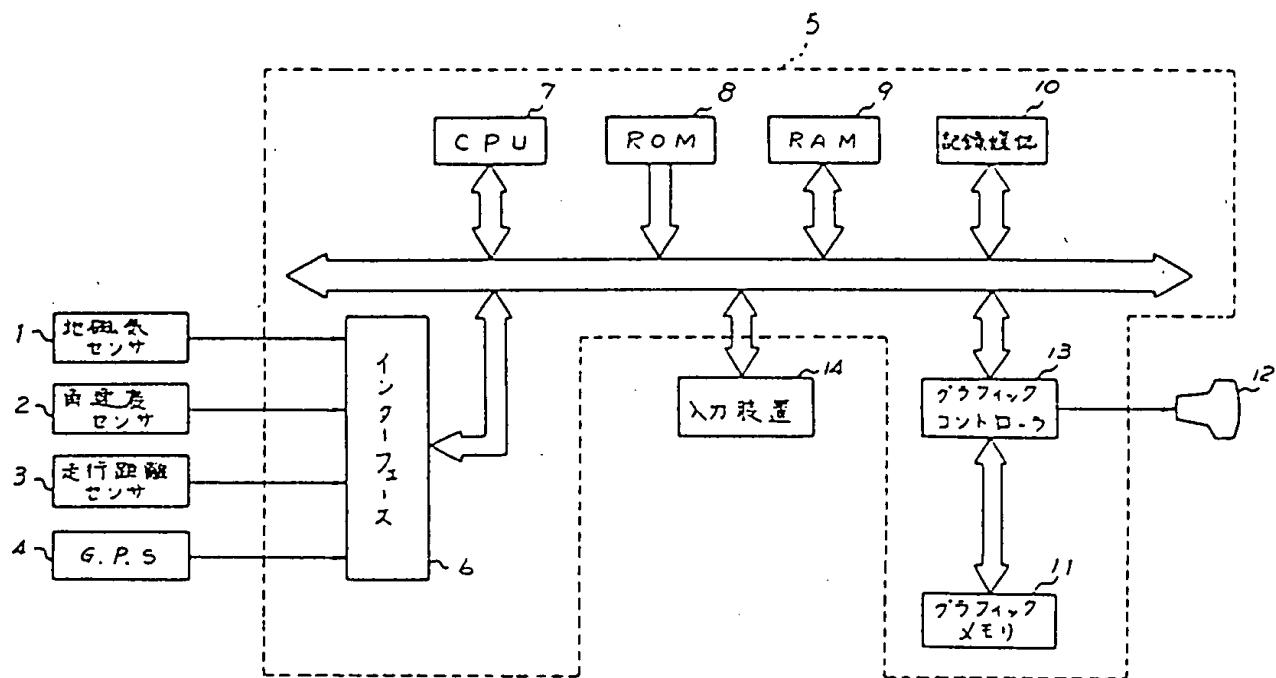
(A)、(B)は第1図における記録媒体に記憶される地図情報のデータ構造を示す図、第14図(A)～(C)はV-RAM上の画面構成を示す図、第15図乃至第18図は第1図におけるCPUによって実行される基本的な手順を示すフローチャート、第19図乃至第21図は方位補正係数θsの求め方を示す図、第22図はCPUによって実行される最近傍移分及び交差点をサーチする手順を示すフローチャート、第23図及び第24図は移分引込みルーチン及び交差点引込みルーチンの手順を示すフローチャート、第25図はCPUによって実行されるタイマー割込みの手順を示すフローチャート、第26図はCPUによって実行されるパターンマッチングルーチンの手順を示すフローチャートである。

主要部分の符号の説明

- 1 …… 地磁気センサ 2 …… 角速度センサ
- 5 …… システムコントローラ
- 7 …… CPU 10 …… 記録媒体
- 12 …… ディスプレイ 14 …… 入力装置

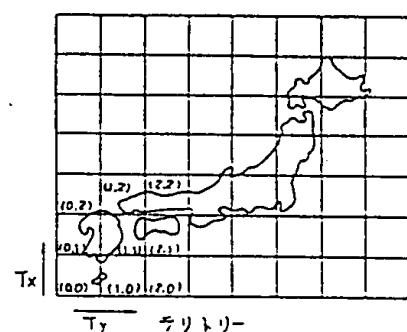


三 / 四



三·2四

(A)



(B)

ナトリトリー ファイル	ナ トリ ト リ ー	4			4		
ID		(10,0)	(11,0)		(i,j)		(k,l,j)

(c)

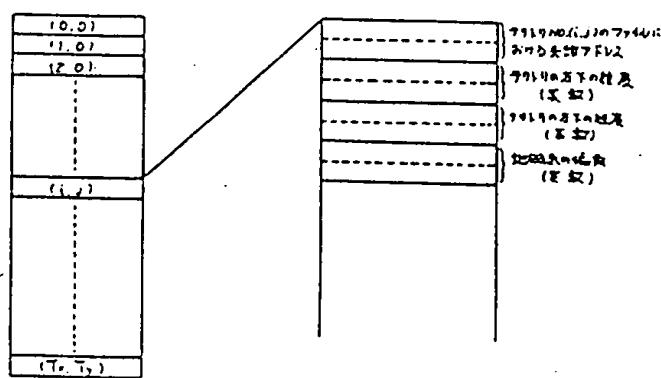
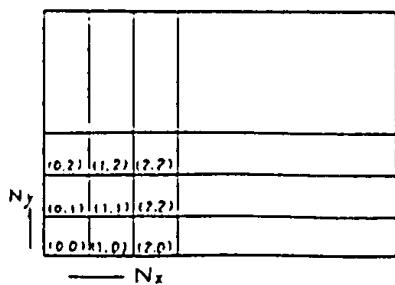
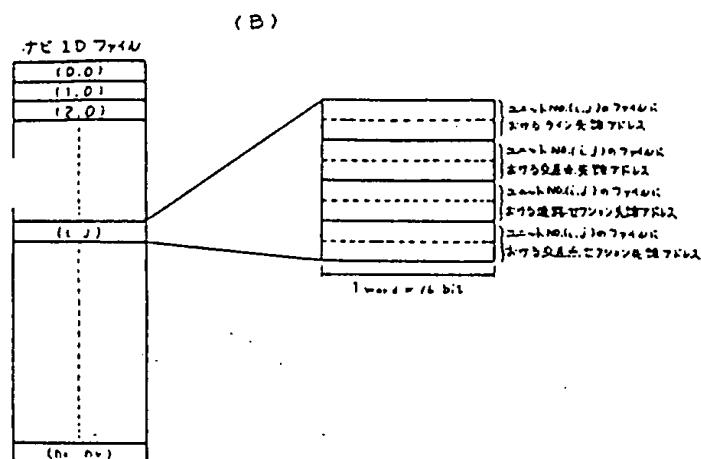
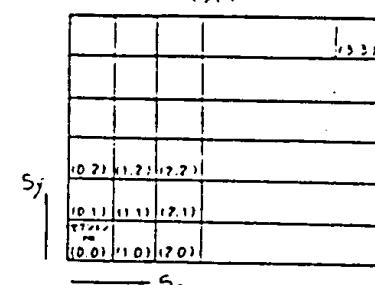


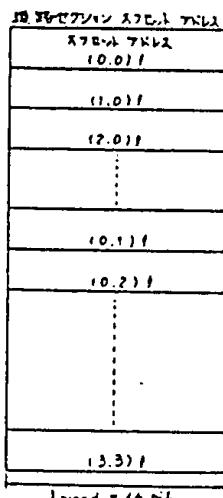
図 4 Z
(A)

$$\begin{cases} N_x = f_x(\text{crnt } x) \\ N_y = f_y(\text{crnt } y) \end{cases}$$

図 5 □
(A)

$$\begin{cases} S_x = g_x(\text{crnt } X) \\ S_y = g_y(\text{crnt } Y) \end{cases}$$

(B)



(C)

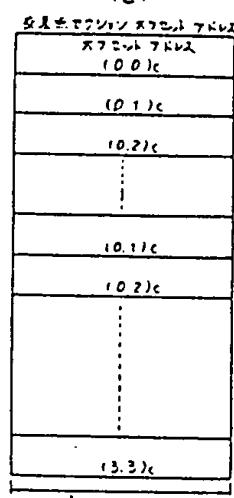
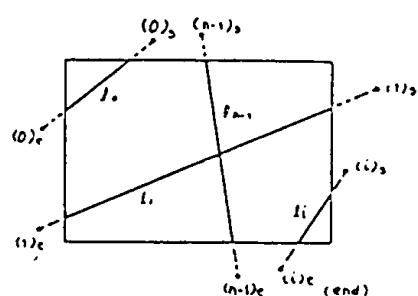


図 6 □

(A)



(B)

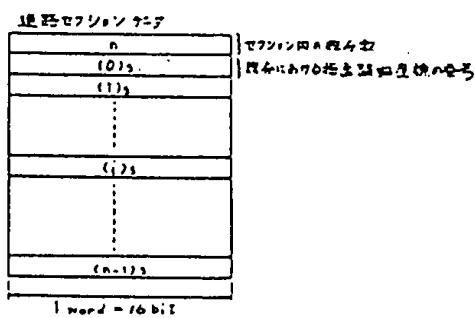
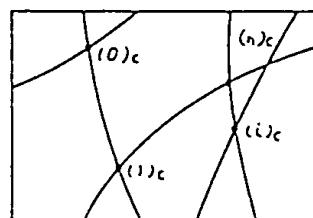


図 7 □

(A)



(B)

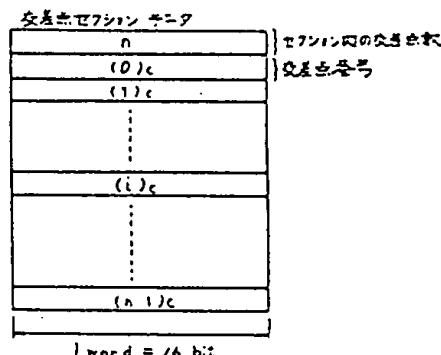
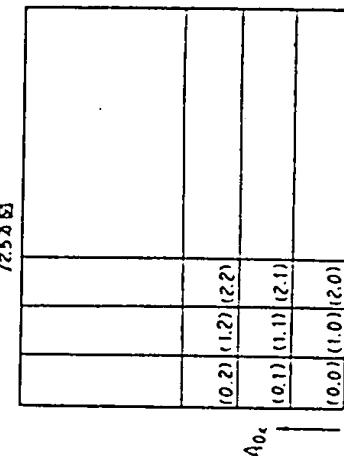


図 8 (A)

 A_{0y}

$$\begin{cases} A = K_{0x} (\text{crnt } x) \\ A = K_{0y} (\text{crnt } y) \end{cases}$$

(B)

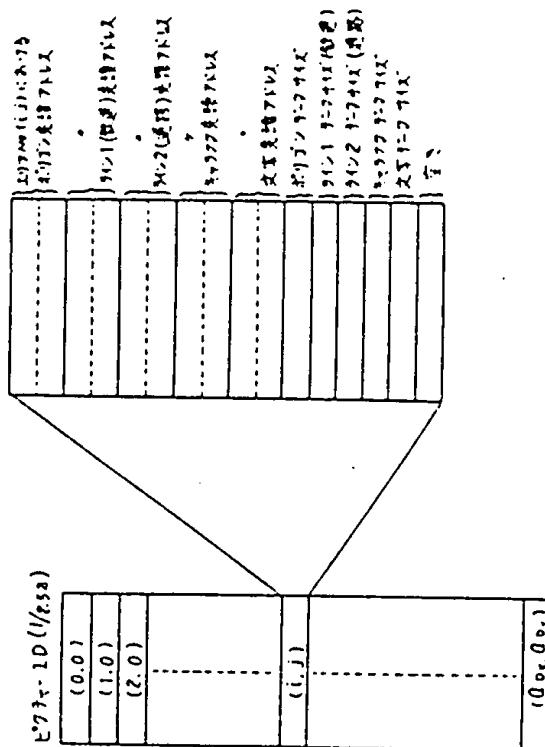
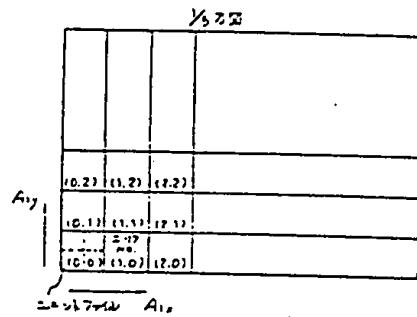
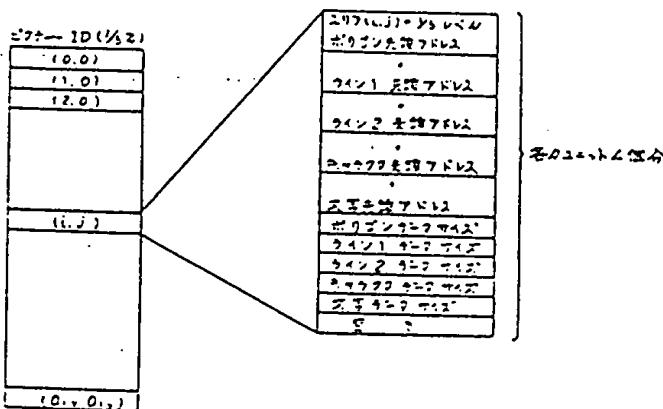
 $\frac{1}{256} \times \frac{1}{256} \times \frac{1}{256}$ (i, j) $(0_0, 0_1)$

図 9 (A)

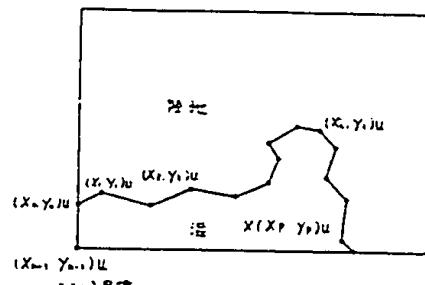
 A_{1y} データアドレス A_{1z}

(B)



各カセット毎

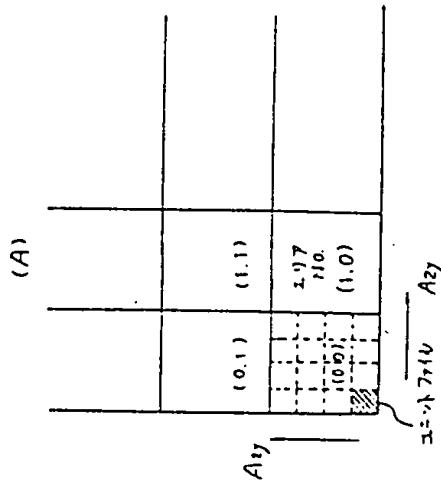
図 11 (A)

 $(X_{n-1}, Y_{n-1})u$

データアドレス

 $(X_n, Y_n)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y_b)u$ $(X_c, Y_c)u$ $(X_d, Y_d)u$ $(X_e, Y_e)u$ $(X_f, Y_f)u$ $(X_g, Y_g)u$ $(X_h, Y_h)u$ $(X_i, Y_i)u$ $(X_j, Y_j)u$ $(X_k, Y_k)u$ $(X_l, Y_l)u$ $(X_m, Y_m)u$ $(X_n, Y_n)u$ $(X_o, Y_o)u$ $(X_p, Y_p)u$ $(X_q, Y_q)u$ $(X_r, Y_r)u$ $(X_s, Y_s)u$ $(X_t, Y_t)u$ $(X_u, Y_u)u$ $(X_v, Y_v)u$ $(X_w, Y_w)u$ $(X_x, Y_x)u$ $(X_y, Y_y)u$ $(X_z, Y_z)u$ $(X_a, Y_a)u$ $(X_b, Y$

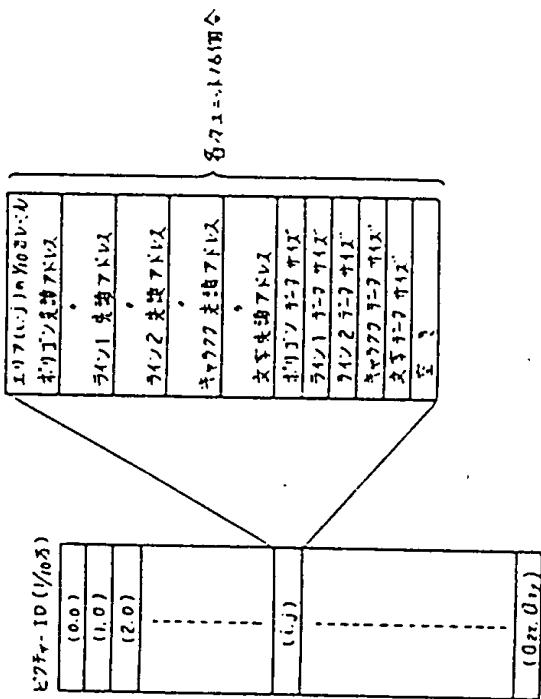
第10図



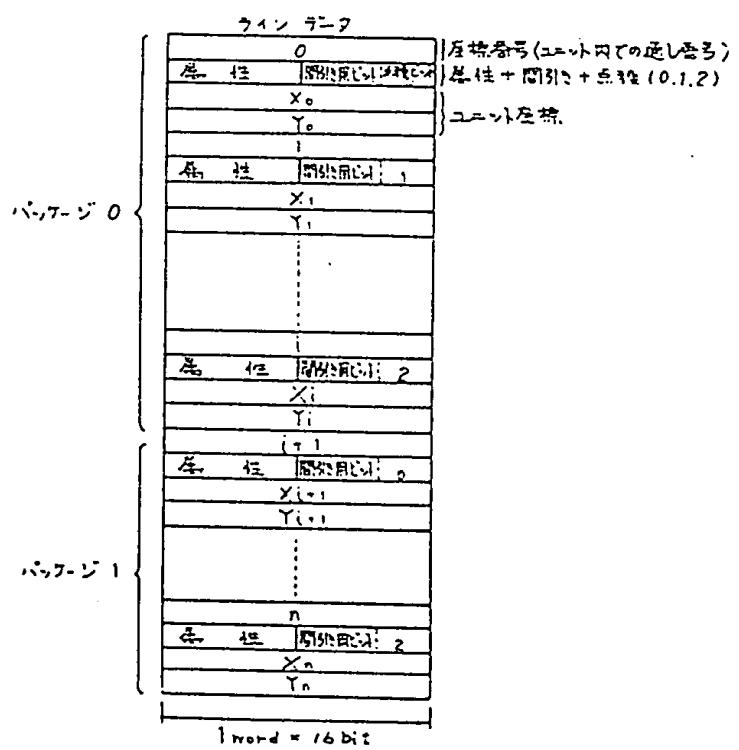
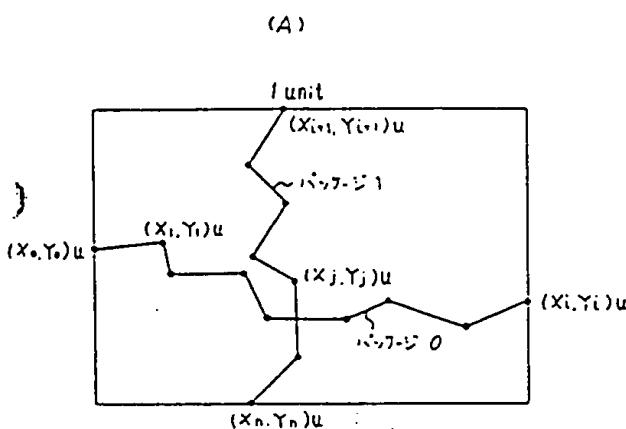
$$A_{2x} = K_{2x} (\text{crnt}_x)$$

$$A_{2y} = K_{2y} (\text{crnt}_y)$$

(B)



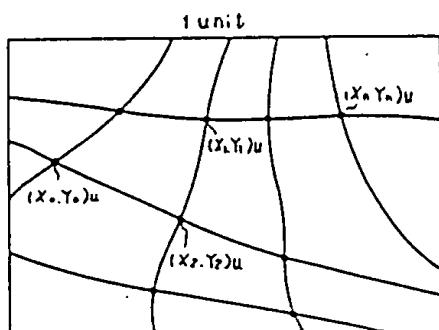
(B)



第13 図

(B)

(A)

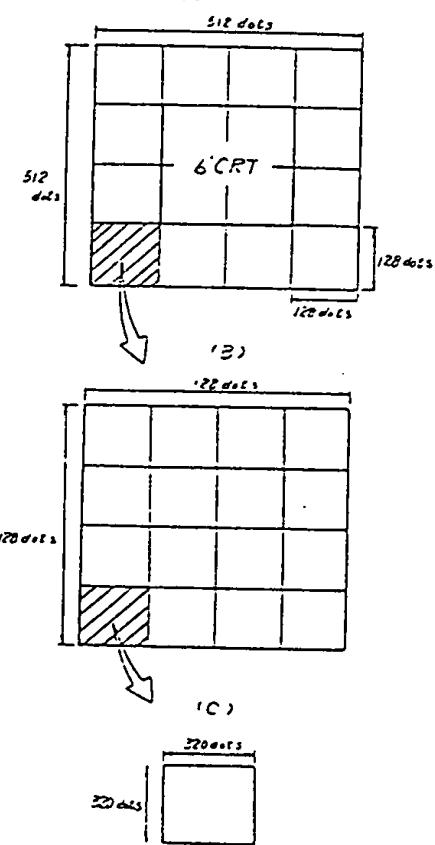


次第点 テーブル

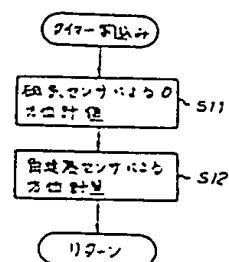
0	交差点番号
属性(レベル)	頂点/辺/内部
X ₀	
Y ₀	
1	属性(レベル)
X ₁	頂点/辺/内部
Y ₁	
	属性(レベル)
X _i	頂点/辺/内部
Y _i	
	属性(レベル)
X _n	頂点/辺/内部
Y _n	

1 word = 16 bit

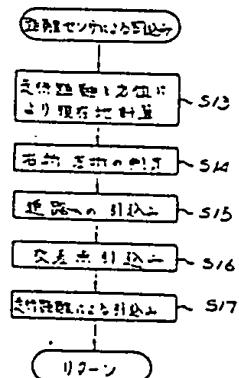
第14 図 (A)



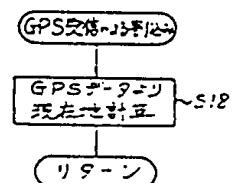
第16 図



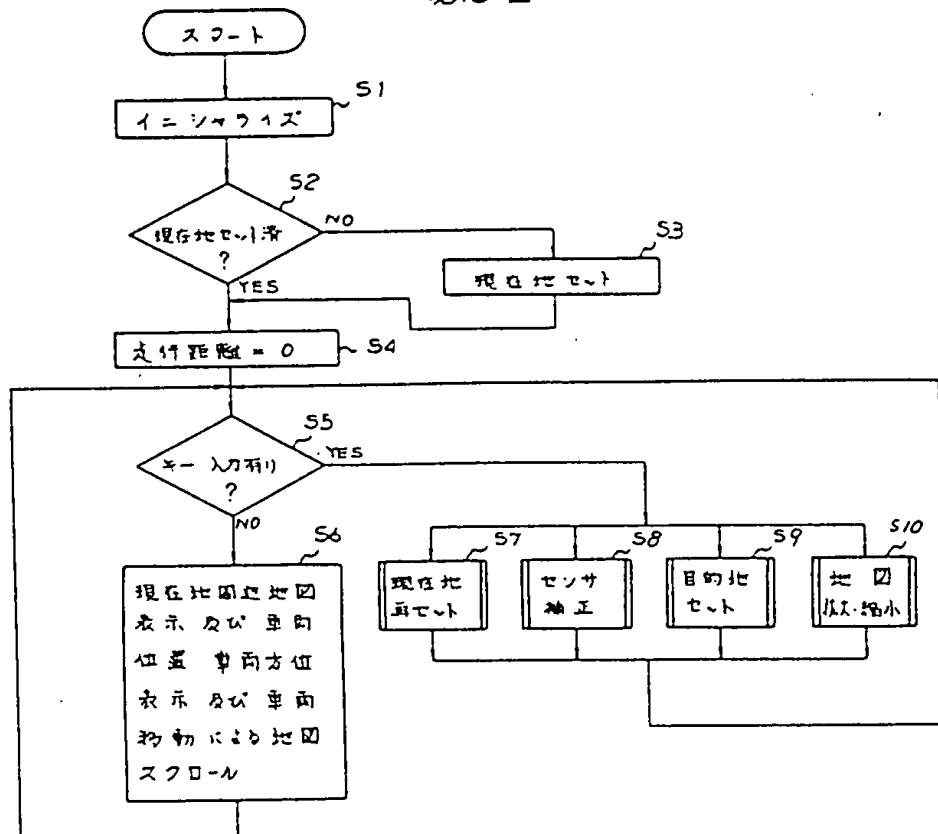
第17 図



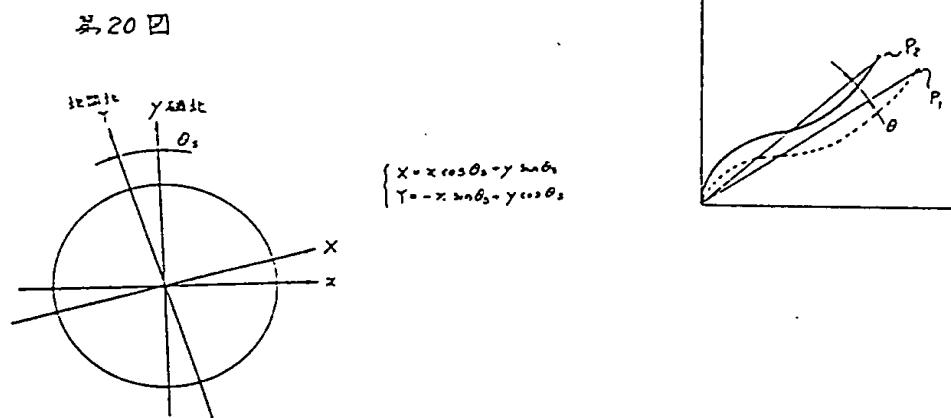
第18 図



第15図



第19図



第20図

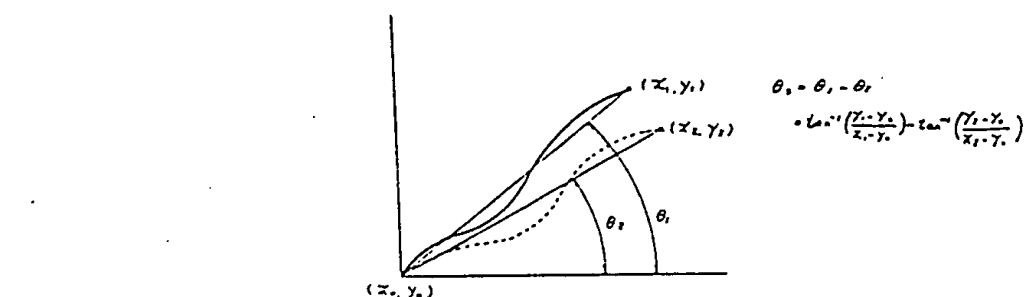


図22

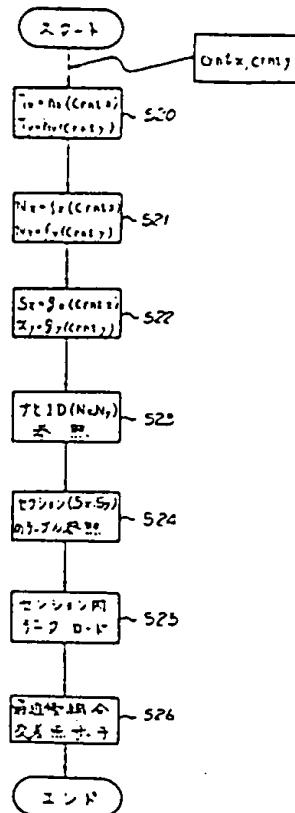


図23

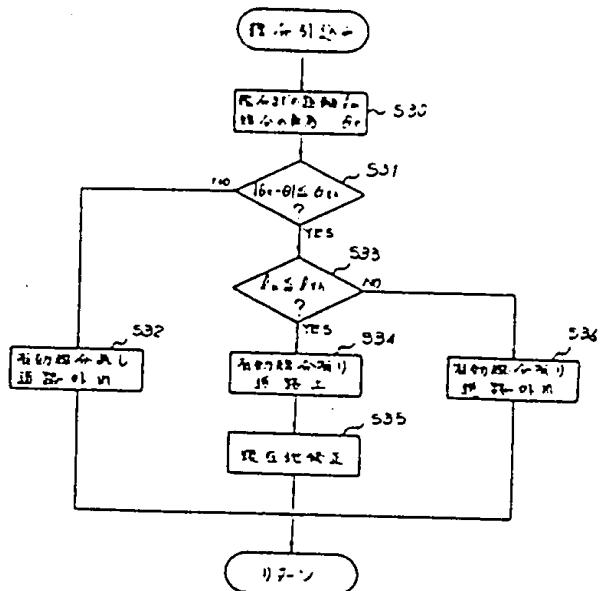


図24

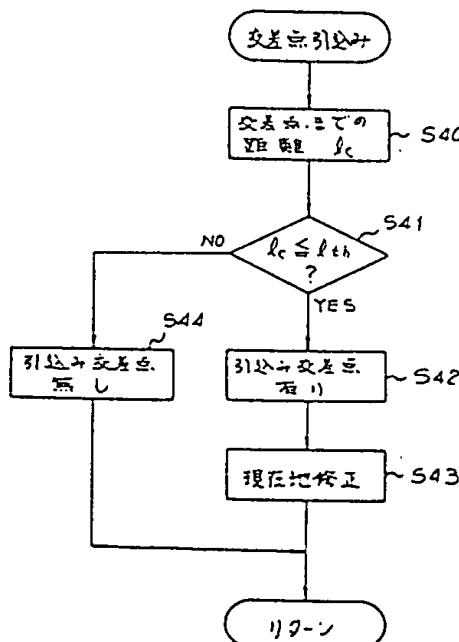


図25

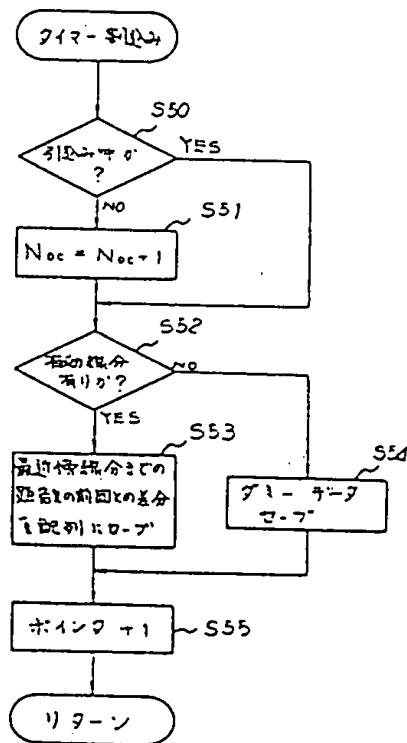


図26

